IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: LANCE W. RUSSELL	Art Unit	: 2143
Serial No.	: 09/895,235	Examiner	: Bilgrami, Asghar H.
Filed	: June 28, 2001	Confirmation No.:	8674
Title	: MIGRATING RECOVERY MODULES IN A DISTRIBUTED COMPUTING ENVIRONMENT		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

II. Related Appeals and Interferences

Appellant is not aware of any related appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims

Claims 1-9, 11-25, and 27-30, which are the subject of this appeal, are pending.

Claims 10 and 26 have been canceled.

Claims 1-9, 11-25, and 27-30 stand rejected.

CERTIFICATE OF TRANSMISSION

I hereby certify that this document is being transmitted to the Patent and Trademark Office via electronic filing on the date shown below.

Dec. 18, 2008
Date of Transmission

/Edouard Garcia, Reg. No. 38,461/
(Signature of person mailing papers)

Edouard Garcia
(Typed or printed name of person mailing papers)

Appellant appeals all rejections of the pending claims 1-9, 11-25, and 27-30.

IV. Status of Amendments

The last amendments filed November 27, 2006, have been entered and acted upon by the Examiner.

No amendments were filed after any of the Office actions dated September 18, 2008, March 20, 2008, and July 31, 2007.

V. Summary of Claimed Subject Matter

In the following Summary, the citations in parentheses are representative of support provided in the application.

A. Independent claim 1

The aspect of the invention defined in independent claim 1 is a system for managing a plurality of distributed nodes of a network (see, e.g., page 4, line 28- page 5, line 1; FIG. 1). The system includes a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes (see, e.g., page 5, lines 6-10; FIG. 1; page 8, lines 14-18; FIG. 3, block 50). Each of the recovery modules is configured to migrate from one network node to another (see, e.g., page 10, lines 11-20; FIG. 5, block 88), determine a respective status of each of the network nodes to which it has migrated (see, e.g., page 9, lines 25-30; FIG. 5, block 80), and initiate a recovery process on ones of the network nodes having one or more failed node processes (see, e.g., page 10, lines 3-5; FIG. 5, blocks 82-84). The recovery modules determine the status of each of the network nodes (see, e.g., page 5, lines 13-15). The network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes (see, e.g., page 8, lines 26-28; FIG. 3, block 52).

B. Independent claim 5

The aspect of the invention defined in independent claim 5 is a system for managing a plurality of distributed nodes of a network (see, e.g., page 4, line 28- page 5, line 1; FIG. 1). The system includes a recovery module (see, e.g., page 8, lines 3-6; FIG. 2, block 20) configured to migrate from one network node to another (see, e.g., page 10, lines 11-20; FIG. 5, block 88), determine a status of a network node (see, e.g., page 9, lines 25-30; FIG. 5, block 80), and initiate a recovery process on a network node having one or more failed node processes (see, e.g., page 10, lines 3-5; FIG. 5, blocks 82-84). The recovery module is configured to determine the status of a network node in accordance with a heartbeat messaging protocol (see, e.g., page 9, line 28 - page 10, line 3).

C. Independent claim 11

The aspect of the invention defined in independent claim 11 is a method for managing a plurality of distributed nodes of a network (see, e.g., page 4, line 28- page 5, line 1; FIG. 1). This method includes (a) on a current one of the network nodes, determining a status of the current network node (see, e.g., page 9, lines 25-30; FIG. 5, block 80); (b) in response to a determination that the current network node has one or more failed node processes, initiating a recovery process on the current network node (see, e.g., page 10, lines 3-5; FIG. 5, blocks 82-84); (c) after initiating the recovery process, migrating from the current network node to a successive one of the network nodes (see, e.g., page 10, lines 11-20; FIG. 5, block 88); and (d) repeating (a), (b), and (c) with the current network node corresponding to the successive network node for each of the nodes in the network (see, e.g., page 5, lines 13-15, and page 3, lines 2-4).

D. Dependent claim 15

Claim 15 depends from claim 11 and recites that the status of a network node is determined in accordance with a heartbeat messaging protocol (page 3, lines 20-25, page 9, line 25 - page 10, line 3, and FIG. 5).

E. Independent claim 20

The aspect of the invention defined in independent claim 20 is a computer program for managing a plurality of distributed nodes of a network (see, e.g., page 4, line 28- page 5, line 1; page 6, lines 13-20; FIG. 2; page 8, lines 14-17; page 10, line 21 - page 11, line 4). The computer program resides on a computer-readable medium and includes computer-readable instructions (see, e.g., page 8, lines 14-17; page 10, line 21 - page 11, line 4). The computer-readable instructions cause a computer to migrate the computer program from one network node to a series of successive network nodes (see, e.g., page 10, lines 11-20; FIG. 5, block 88). The computer-readable instructions cause a computer to determine a status of a current one of the network nodes to which the computer program has migrated (see, e.g., page 9, lines 25-30; FIG. 5, block 80). The computer-readable instructions cause a computer to initiate a recovery process on the current network node in response to a determination that the current network has one or more failed node processes (see, e.g., page 10, lines 3-5; FIG. 5, blocks 82-84). The computer-readable instructions cause a computer to migrate from the current network node to a successive one of the network nodes after initiating the recovery process on the current network node (see, e.g., page 10, lines 11-20; FIG. 5, block 88).

F. Dependent claim 27

Claim 27 depends from claim 1 and recites that the network management module determines a number of the recovery modules needed to achieve a specified network monitoring service level, and launches the determined number of recovery modules into the network to achieve the specified network monitoring service level (see, e.g., page 8, line 24 - page 9, line 5; FIG. 3, blocks 50-54).

G. Dependent claim 28

Claim 28 depends from claim 1 and recites that the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and launches the recovery modules into the network by

transmitting respective ones of the recovery modules to the identified target network nodes (see, e.g., page 8, lines 21-24; FIG. 3, block 50).

H. Dependent claim 29

Claim 29 depends from claim 11 and recites: determining a number of the recovery modules needed to achieve a specified network monitoring service level (see, e.g., page 8, line 24 - page 9, line 5; FIG. 3, blocks 50-54); statistically identifying target ones of the network nodes to achieve a specified confidence level of network monitoring reliability (see, e.g., page 8, lines 21-24; FIG. 3, block 50); and transmitting the determined number of the recovery modules to the identified target network nodes (see, e.g., page 8, line 24 - page 9, line 5; FIG. 3, blocks 50-54).

VI. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1-4, 6-9, 11-14, 16-25, and 30 stand rejected under 35 U.S.C § 102(e) over Turek (U.S. 6,460,070).

B. Claims 5 and 15 stand rejected under 35 U.S.C § 103(a) over Turek (U.S. 6,460,070) in view of Sreenivasan (U.S. 2002/0049845).

C. Claims 27-29 stand rejected under 35 U.S.C. § 103(a) over Turek (U.S. 6,460,070) in view of Douik (U.S. 6,012,152).

VII. Argument

A. Rejection of claims 1-4, 6-9, 11-14, 16-25, and 30 under 35 U.S.C. § 102(e) over Turek

The Examiner has rejected claims 1-4, 6-9, 11-14, 16-25, and 30 under 35 U.S.C § 102(e) over Turek (U.S. 6,460,070).

1. Applicable standards for sustaining a rejection under 35 U.S.C. § 102(e)

The relevant part of 35 U.S.C. § 102(e) states that a person shall be entitled to an invention, unless - "the invention was described in -- (1) an application for patent published under section 122(b), by another filed in the United States before the invention by the applicant for patent..." Anticipation under 35 U.S.C. § 102(e) requires that each and every element of the claimed invention be present, either expressly or inherently, in a single prior art reference. EMI Group N. Am., Inc., v. Cypress Semiconductor Corp., 268 F.3d 1342, 1350 (Fed. Cir. 2001). Anticipation must be proved by substantial evidence. In re Crish, 393 F.3d 1253, 73 USPQ2d 1364 (Fed. Cir. 2004).

2. Independent claim 1

a. Introduction

Independent claim 1 recites:

> 1. A system for managing a plurality of distributed nodes of a network, comprising:
>
> a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes;
>
> wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes, and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes.

As explained in detail below, the rejection of independent claim 1 under 35 U.S.C. § 102(e) over Turek should be withdrawn because Turek neither expressly nor inherently discloses each and every element of the invention defined by the claim.

b. The Examiner's position

In support of the rejection of independent claim 1, the Examiner has stated that (see § 7, pages 4-5 of the Office action dated September 18, 2008; emphasis added):

> As per claims 1, 11, 19 & 20 Turek-Sreenivasan [sic] disclosed a method for managing a plurality of distributed nodes of a network, comprising: a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on failed ones of the network nodes.(col.3, lines 48-64, col.1, lines 59-62, 65-67, col.2, lines 22-26, col.2, lines 1-3, col.2, lines 22-26 & col.5, lines 32-60), having one or more failed node processes, the recovery modules determine the status of each of the network nodes, and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of the network nodes (col.7, lines 58-67 & col.8, lines 1-9) after initiating the recovery process, migrating from the current node to a successive one of the network node (col.5, lines 32-60, col.7, lines 58-67 & col.8, lines 1-65), and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes (col.7, lines 58-67, col. 8, lines 1-9 & col.8, lines 39-58).

c. Appellant's rebuttal: Turek does not disclose each and every element of claim 1

i. Turek does not disclose a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes

Turek does not disclose or suggest a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes.

As explained in detail below, Turek's system does not launch migratory recovery modules into a network to monitor status of each of the network nodes, wherein the recovery modules determine the status of each of the network nodes and the network management module

monitors transmissions that are received from the recovery modules to provide periodic monitoring of each of the network nodes.

(1) Turek does not disclose a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes

(a) Turek's disclosure

In accordance with Turek's disclosure, the mobile software agents are deployed by the dispatch mechanism 15 in the management server 14 only in response to either a report of a "network 'fault', alarm or other such trigger" (col. 7, lines 3-4) or a "request for maintenance in some non-specified area of the network" (col. 7, line 7). The dispatch mechanism 15 deploys a selected one of the software agents to the particular location of a fault or a particular area of a network where the fault is likely to have occurred (see, e.g., col. 7, lines 1-57). If the initial given node location does not contain the specific fault for which the software agent was deployed, the software agent identifies "a subset of nodes (associated with the given node) that remain candidates for locating the error" (col. 8, lines 31-32). Once the particular fault is located and diagnosed, the software agent attempts to fix the problem (see col. 9, lines 21-22). "If unable to effect repairs, the agent will, at a minimum, report back with the diagnosis to a user interface of the dispatch mechanism" (col. 9, lines 28-30). Turek does not teach or suggest anything that that would have led one skilled in the art at the time the invention was made to believe that the software agent migrates to any other nodes after attempting to "effect repairs" and reporting back with the diagnosis. To the contrary, in accordance with Turek's teachings, each of the mobile software agents is deployed to diagnose and, if possible correct, only one particular network fault (see, e.g., col. 5, lines 43-60). Therefore, there is no apparent need for any of Turek's software agents to migrate from the node that contains the particular network fault that the software agent was deployed to diagnose and correct.

Thus, one skilled in the art at the time the invention was made would not have had any basis for believing that Turek's system launches migratory recovery modules into a network to monitor status of each of the network nodes, as recited in claim 1. In accordance with its ordinary and accustomed meaning, the verb "monitor" means "to watch, keep track of, or check

usu. for a special purpose" (Merriam-Webster's Collegiate Dictionary, Tenth Edition (1995). The dispatch mechanism 15 does not launch migratory recovery modules into the network to monitor status of each of the network nodes. Instead, the dispatch mechanism 15 deploys the software agents only after a network fault already has been determined by the management server 14 (see, e.g., col. 7, lines 3-4) or after receiving a "request for maintenance in some non-specified area of the network" (col. 7, line 7), and the dispatched agents cease migrating after reaching their respective target nodes.

In addition, one skilled in the art at the time the invention was made would not have had any basis for believing that Turek's software agents determine the status of each of the nodes of a network. To the contrary, based on Turek's teaching, such a person would have recognized that the software agents are designed to recursively narrow the search for the particular network nodes for which they were respectively deployed and, after arriving at the target network nodes, the software agents attempt to effect repairs and report back diagnoses; the dispatched agents cease migrating after reaching their respective target nodes. That is, the software agents are not configured to determine the status of each of the network nodes, as recited in claim 1. Moreover, Turek only teaches that each of the software agents is configured to determine whether a particular event originated from a node (see col. 2, lines 49-53). Turek does not teach that these agents are configured to determine the status of their respective target nodes. Consequently, the dispatch mechanism 15 is not configured to provide periodic monitoring of the status of each of the network nodes, as recited in claim 1.

(b) The cited sections of Turek's disclosure do not support the Examiner's position

The Examiner has pointed to col. 3, lines 48-64, col.1, lines 59-62, 65-67, col.2, lines 22-26, col.2, lines 1-3, col.2, lines 22-26 & col.5, lines 32-60 in support of the position that Turek discloses "a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the

status of each of the network nodes" (see § 7 on page 4 of the Office action dated September 18, 2008). As explained in detail below, however, the cited sections of Turek's disclosure, however, do not support the Examiner's characterization of Turek's teachings.

Col.3, lines 48-64:

Col.3, lines 48-64 recites:

> Referring now to FIG. 1, the invention is preferably implemented in a large distributed computer environment 10 comprising up to thousands of "nodes." The nodes will typically be geographically dispersed and the overall environment is "managed" in a distributed manner. Preferably, the managed environment (ME) is logically broken down into a series of loosely-connected managed regions (MR) 12, each with its own management server 14 for managing local resources with the MR. The network typically will include other servers (not shown) for carrying out other distributed network functions. These include name servers, security servers, file servers, threads servers, time servers and the like. Multiple servers 14 coordinate activities across the enterprise and permit remote site management and operation. Each server 14 serves a number of gateway machines 16, each of which in turn support a plurality of endpoints 18. The server 14 coordinates all activity within the MR using a terminal node manager 20.

This disclosure does not teach or suggest anything about launching migratory recovery modules, much less anything about "a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes."

Col. 1, lines 59-62, 65-67:

Col. 1, lines 59-62, 65-67 recites:

> It would be a significant advantage to provide some automatic means of diagnosing and correcting network problems in this type of computer environment. The present invention addresses this important problem.
>
> ...
>
> It is a primary object of this invention to automatically diagnose faults or other events that occur in a large, distributed computer network.

This disclosure does not teach or suggest anything about launching migratory recovery modules, much less anything about "a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes."

It is noted that diagnosing and correcting network problems does not constitute monitoring the status of each of nodes in a network.

Col. 2, lines 1-3:

Col. 2, lines 1-3 recites that "It is another primary object of this invention to deploy a software "agent" into a distributed computer network environment to diagnose and, if possible, correct a fault."

This disclosure does not teach or suggest "a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes."

It is noted that diagnosing and correcting network problems does not constitute monitoring the status of each of nodes in a network.

Col. 2, lines 22-26:

Col. 2, lines 22-26 recites:

> Yet another object of the present invention is to collect information about network conditions as mobile software agents are dispatched and migrated throughout a large computer network to correct network faults, wherein such information is then useful in diagnosing new faults.

This disclosure does not teach or suggest "a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes."

It is noted that collecting from mobile software agents information that is useful in diagnosing new faults does not constitute monitoring the status of each of nodes in a network.

Col. 5, lines 32-60

Col. 5, lines 32-60 recites:

> A preferred embodiment of the present invention is implemented in the enterprise environment illustrated above. In this embodiment, a set of "software agents" are available at a central location (e.g., manager 14) or at a plurality of locations (e.g., the gateways 16) in the o network where network errors are reported. The software agents are "mobile" in the sense that the agents are dispatched (as will be described below) from a dispatch mechanism and then migrate throughout the network environment. Generally, the mobile software agents traverse the network to diagnose and, if possible, to correct a network fault.
>
> Thus, when a network error or "fault" is reported whose cause and location are not apparent or readily ascertainable, an appropriate agent is identified and dispatched to determine this information. Preferably, the agent is dispatched to the actual node in the network at which the fault condition occurs. As will be seen, the

> particular error, as well as other associated events, generally provide a "clue" or clues regarding the network location to which the agent should be sent, as well as the type of agent to send. If the agent does not find the fault at the initial location to be examined, the agent then migrates through the network to locate the error. The agent preferably chooses its path through the network based on the information received at the dispatching location, as well as information gleaned from each examined location. As will be seen, the particular "path" typically varies as the software agent migrates through the network because information gleaned from a particular node may redirect the agent in some given manner.

This disclosure does not teach or suggest "a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes." Instead, the cited disclosure merely teaches that each of the mobile software agents is deployed to diagnose and, if possible correct, a particular network fault (see, e.g., col. 5, lines 43-60).

It is noted that diagnosing and correcting network problems does not constitute monitoring the status of each of nodes in a network.

(2) Conclusion

For the reasons explained above, neither Turek nor Sreenivasan discloses or suggests a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes.

ii. Turek also does not disclose a network management module that "monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes"

In the Office action dated March 20, 2008, the Examiner had acknowledged that "Turek did not explicitly disclose the recovery module (software agents) periodically sending network node status" (see page 6, lines 8-9, of the Office action dated March 20, 2008). Yet, now in the

Office action dated September 18, 2008, the Examiner has taken the position that Turek discloses "the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of the network nodes" in col. 7, lines 58-67 and col. 8, lines 1-9 (see § 7 on page 4 of the Office action dated September 18, 2008).[1] As explained in detail below, however, the cited sections of Turek's disclosure, however, do not support the Examiner's characterization of Turek's teachings.

Col.7, lines 58-67 - col. 8, lines 1-9:

Col.7, lines 58-67 - col. 8, lines 1-9 recites (emphasis added):

> The method continues at step 46 with the software agent being deployed into the network. At step 48, the software agent migrates though the network. A test is then done at step 50 to determine whether the software agent has located the fault. If the outcome of the test at step 50 is negative, the routine cycles. If, however, the outcome of the test at step 50 indicates that the software agent has arrived at the fault location, the routine continues at step 52. At this step, software agent (either alone, or together with some functionality provided by the runtime engine already resident on the node) diagnoses the fault. At step 54, a test is done to determine whether the software agent (either alone, or together with some functionality provided by the runtime engine) can correct the problem. If the outcome of the test at step 54 is positive, the routine continues at step 56 and the problem is rectified. At step 58, information about the problem and the corrective action that as undertaken are reported back to the dispatch mechanism and stored in the database for future use.

This disclosure does not disclose that "the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes," as recited in claim 1.

---

[1] It is noted that the Examiner has misquoted the language of claim 1. In particular, claim 1 does not recite "...to provide periodic monitoring of the status of the network nodes"; instead, claim 1 recites "...to provide periodic monitoring of the status of each of the network nodes" (emphasis added).

In pertinent part col. 7, line 58 - col. 8, line 9, discloses that a test is done and the outcome of that test indicates whether or not the software agent has arrived at the fault location. Turek does not provide any details about which component performs the test that is "done" at step 50 of FIG. 4, nor does Turek reveal anything about the type of information that is input into the test nor the nature of the outcome of the test that indicates whether or not the software agent has arranged at the fault location. Without such details there is no basis on which one skilled in the art reasonably could conclude that Turek's dispatch mechanism monitors transmissions that are received from the software agent to provide periodic monitoring of the status of the network nodes. For example, one reasonably could imagine an embodiment in which the software agent transmits to the display agent a message that indicates whether or not the software agent has arrived at the targeted fault location. Such a message reveals the status of the software agent, not the node.

Moreover, claim 1 recites "the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes" (emphasis added). In accordance with Turek's disclosure, the dispatch mechanism 15 deploys the software agents only after a network fault already has been determined by the management server 14 (see, e.g., col. 7, lines 3-4) or after receiving a "request for maintenance in some non-specified area of the network" (col. 7, line 7), and the dispatched agents cease migrating after reaching their respective target nodes. Thus, a particular node can be expected to be visited only once by the software agents that are deployed by the dispatch mechanism. Accordingly, such a deployment of software agents could not possibly provide periodic monitoring of the status of each of the network nodes.

For the reasons explained above, the cited sections of Turek's disclosure do not support the Examiner's assertion that Turek discloses "the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes," as recited in claim 1

iii. Conclusion

As explained above, Turek's system does not launch migratory recovery modules into a network to monitor status of each of the network nodes, wherein the recovery modules determine the status of each of the network nodes and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of each of the network nodes, as recited in claim 1.

For at least these reasons, the rejection of independent claim 1 under 35 U.S.C. § 102(e) over Turek should be withdrawn.

3. Claims 2-4, 6-9, 21-25, and 30

Each of claims 2-4, 6-9, 21-25, and 30 incorporates the features of independent claim 1 and therefore is patentable over Turek for at least the same reasons explained above.

4. Independent claim 11

Independent claim 11 recites:

> 11. A method for managing a plurality of distributed nodes of a network, comprising:
>
> (a) on a current one of the network nodes, determining a status of the current network node;
>
> (b) in response to a determination that the current network has one or more failed node processes, initiating a recovery process on the current network node;
>
> (c) after initiating the recovery process, migrating from the current network node to a successive one of the network nodes; and
>
> (d) repeating (a), (b), and (c) with the current network node corresponding to the successive network node for each of the nodes in the network.

In support of the rejection of independent claim 11, the Examiner has stated that (see § 7 on pages 4-5 of the Office action dated September 18, 2008; emphasis added):

> As per claims 1, 11, 19 & 20 Turek-Sreenivasan [sic] disclosed a method for managing a plurality of distributed nodes of a network, comprising: a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes; wherein each of the recovery modules is configured to migrate from one network to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on failed ones of the network nodes.(col.3, lines 48-64, col.1, lines 59-62, 65-67, col.2, lines 22-26, col.2, lines 1-3, col.2, lines 22-26 & col.5, lines 32-60), having one or more failed node processes, the recovery modules determine the status of each of the network nodes, and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of the network nodes (col.7, lines 58-67 & col.8, lines 1-9) <u>after initiating the recovery process, migrating from the current node to a successive one of the network node (col.5, lines 32-60, col.7, lines 58-67 & col.8, lines 1-65)</u>, and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes (col.7, lines 58-67, col. 8, lines 1-9 & col.8, lines 39-58).

Contrary to the Examiner's assertion, however, Turek's software agents do not perform any of elements (c) and (d) of independent claim 11.

In accordance with Turek's teachings, the mobile software agents do not migrate from a current network node to a successive one of the network nodes after initiating a recovery process on the current network node. Instead, after initiating a recovery process, Turek's mobile software agents merely report the problem and the corrective action that was taken to the dispatch mechanism 15 (see col. 8, lines 6-9; FIG. 4). Turek does not teach or suggest anything that that would have led one skilled in the art at the time the invention was made to believe that the software agent migrates to any other nodes after attempting to "effect repairs" and reporting back with the diagnosis. Indeed, in accordance with Turek's teachings, each of the mobile software agents is deployed to diagnose and, if possible, correct only one particular network

fault. Therefore, there is no apparent need for any of Turek's software agents to migrate from the node that contains the particular network fault that the software agent was deployed to diagnose and correct.

Thus, Turek does not disclose either element (c) or element (d) of claim 11. For at least this reason, the Examiner's rejection of independent claim 11 under 35 U.S.C. § 102(e) over Turek should be withdrawn.

In the Office action dated March 20, 2008, the Examiner had stated that (see §32 on pages 13-14; original emphasis):

> As to applicant argument with respect to claim language "after initiating the recovery process, migrating from the current network node to a successive node" in claims 11 & 20. The examiner gave 112 first paragraph rejection on August 25-2006 for lack of indication in the applicant's specification regarding this limitation "after initiating the recovery process, migrating from the current network node to a successive node". Applicant's response on December-1-2006 was "It is well-settled, however, that the specification need not contain a literal transcription of the claim language defining the invention in order to satisfy the written description requirement. Instead, the application need only reasonably convey the claimed subject matter to a person in the ordinary skill in the art. In accordance with MPEP 2164.II.A.3(b).
>
> Therefore Examiner is applying the same rationale that the disclosure of the applied reference(s) need not contain a literal transcription of the claim language defining the invention. Instead, the reference(s) need only reasonably convey the claimed subject matter to a person in the ordinary skill in the art.

In the Appeal Brief dated June 17, 2008, Appellant explained that the Examiner's statement quoted above does not address in any way appellant's point that Turek's mobile software agents do not migrate from a current network node to a successive one of the network nodes after initiating a recovery process on the current network node. Instead, after initiating a recovery process, Turek's mobile software agents merely report the problem and the corrective action that was taken to the dispatch mechanism 15 (see col. 8, lines 6-9; FIG. 4). Furthermore, there is no apparent need for Turek's software agents to migrate from the node that contains the particular network fault that the software agent was deployed to diagnose and correct because

each of the mobile software agents is deployed to diagnose and, if possible correct, only one particular network fault (see, e.g., col. 5, lines 43-60, and col. 7, line 58 - col. 8, line 17). This point has been made repeatedly by the Appellant, yet the Examiner consistently has failed to address it.

5. Claims 12-14 and 16-19

Each of claims 12-14 and 16-19 incorporates the features of independent claim 11 and therefore is patentable over Turek for at least the same reasons explained above.

5. Independent claim 20

Claim 20 recites that the computer program comprises computer-readable instructions for causing a computer to perform operations comprising:

> migrating the computer program from one network node to a series of successive network nodes;
>
> determining a status of a current one of the network nodes to which the computer program has migrated;
>
> in response to a determination that the current network has one or more failed node processes, initiating a recovery process on the current network node; and
>
> after initiating the recovery process on the current network node, migrating from the current network node to a successive one of the network nodes.

Claim 20 is patentable over Turek in view of Sreenivasan for at least the same reasons explained above in connection with independent claim 11. Accordingly, the Examiner's rejection of independent claim 20 under 35 U.S.C. § 102(e) over Turek should be withdrawn.

B. Rejection of claims 5 and 15 under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan

The Examiner has rejected claims 5 and 15 under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan (U.S. 2002/0049845).

1. Applicable standards for sustaining a rejection under 35 U.S.C. § 103(a)

"A patent may not be obtained ... if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." 35 U.S.C. §103(a).

In an appeal involving a rejection under 35 U.S.C. § 103, an examiner bears the initial burden of establishing *prima facie* obviousness. See In re Rijckaert, 9 F.3d 1531, 1532, 28 USPQ2d 1955, 1956 (Fed. Cir. 1993). To support a *prima facie* conclusion of obviousness, the prior art must disclose or suggest all the limitations of the claimed invention.[2] See In re Lowry, 32 F.3d 1579, 1582, 32 USPQ2d 1 031, 1034 (Fed. Cir. 1994). If the examiner has established a *prima facie* case of obviousness, the burden of going forward then shifts to the applicant to overcome the *prima facie* case with argument and/or evidence. Obviousness, is then determined on the basis of the evidence as a whole and the relative persuasiveness of the arguments. This inquiry requires (a) determining the scope and contents of the prior art; (b) ascertaining the differences between the prior art and the claims in issue; (c) resolving the level of ordinary skill in the pertinent art; and (d) evaluating evidence of secondary consideration. See KSR Int'l Co. v. Teleflex Inc., No. 127 S. Ct. 1727, 1728 (2007) (citing Graham v. John Deere, 383 U.S. I, 17-18,

[2] The U.S. Patent and Trademark Office has set forth the following definition of the requirements for establishing a *prima facie* case of unpatentability (37 CFR § 1.56(b)(ii)):

> A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

148 USPQ 459, 467 (1966)). If all claim limitations are found in a number of prior art references, the fact finder must determine whether there was an apparent reason to combine the known elements in the fashion claimed. See KSR, 1741. This analysis should be made explicit. KSR at 1741 (citing In re Kahn, 441 F. 3d 977, 988 (Fed. Cir. 2006): "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness").

2. Independent claim 5

a. Introduction

Independent claim 5 recites:

> 5. A system for managing a plurality of distributed nodes of a network, comprising:
>
> a recovery module configured to migrate from one network node to another, determine a status of a network node, and initiate a recovery process on a network node having one or more failed node processes, wherein the recovery module is configured to determine the status of a network node in accordance with a heartbeat messaging protocol.

As explained in detail below, the rejection of independent claim 5 under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan should be withdrawn because (1) the Examiner has not shown that the cited references disclose each and every element of the claim, and (2) one skilled in the art at the time the invention was made would not have had any apparent reason to modify the teachings of Turek in the manner proposed by the Examiner.

b. The Examiner's position

In support of the rejection of claim 5, the Examiner has stated that (see § 3, pages 2-3 of the Office action dated September 18, 2008; emphasis added):

As per claims 5 Turek disclosed a system for managing a plurality of distributed nodes of a network, comprising: a recovery modules configured to migrate from one network node to another, determine a status of a network, and initiate a recovery process on a network node having one or more failed node processes (col.2, lines 6567 & col.2, lines 1-46) wherein the recovery module is configured to determine the status of a network node in accordance with a heartbeat messaging protocol (col.2, lines 22-46). Although Turek disclosed software agent (module) providing network status information to the management module. However Turek did not specifically mentioned agent using a "heartbeat messaging protocol" to determine the status of a network node. In the same field of endeavor Sreenivasan disclosed daemon (module or software agent) sending "I am alive message" (heartbeat messaging protocol) to determine the status of a network node (paragraphs 78, 111 & 112).

It would have been obvious to one in the ordinary skill in the art at the time the invention was made to have incorporated the functionality of daemon (module or software agent) sending "I am alive message" (heartbeat messaging protocol) to determine the status of a network node as disclosed by Sreenivasan in the a system for managing a plurality of distributed nodes of a network as disclosed by Turek in order to make the managing system more reliable and responsive resulting in determining accurate diagnosis and status of the network nodes.

c. Appellant's rebuttal: the cited references do not disclose each and every element of claim 5

Claim 5 recites that "the recovery module is configured to determine the status of a network node in accordance with a heartbeat messaging protocol." Thus, under a proper construction of claim 5, the term "status" must be construed as a status of the network node that is determinable in accordance with a heartbeat messaging protocol.

Heartbeat messaging is a well-known feature of networks in accordance with which a "heartbeat" message is sent regularly from one node to another node merely to detect failed applications or failed nodes (see, e.g., Forbes, U.S. 6,728,896, col. 4, lines 27-33, and col. 9, lines 2-14; cited by the Examiner). Such heartbeat messages indicate the statement "I'm here, are you here?" (see, e.g., col. 9, lines 2-14, of Forbes, U.S. 6,728,896; cited by the Examiner).

As explained above, Turek does not disclose or suggest a recovery module that is configured to determine the status of a network node in accordance with a heartbeat messaging protocol. Instead, Turek discloses that network errors or faults are determined by the remote management server 14 (see, e.g., col. 5, lines 31-60). When a network error or fault is determined, the management node 14 deploys the mobile software agents to diagnose and, if possible, correct the network error or fault (see, e.g., col. 5, lines 37-42). Given the limited information provided by heartbeat messages, they cannot be used to "diagnose and, if possible, correct a network fault" as required of the mobile software agents (see Turek, col. 5, lines 41-42). Instead, the mobile software agents perform these tasks by performing tests at the nodes to which the software agents were deployed by the management server 14 (see col. 7, line 58 - col. 8, line 9). Thus, Turek fails to disclose or suggest software agents that determine the "status" of a network node, where the "status" is determinable in accordance with a heartbeat messaging protocol.

Contrary to the Examiner's position, however, Sreenivasan does not disclose anything about recovery modules of the type disclosed in Turek, much less anything about such modules that are "configured to determine the status of a network node in accordance with a heartbeat messaging protocol."

In paragraph 78, Sreenivasan discloses (emphasis added):

> In the embodiment shown in FIG. 1, the two server systems 12 are connected to both a public network 14 and a private network 18. Clients 16 use public network 14 to access services from the cluster. Software running on each server 12 use private network 18 to exchange heartbeat and other control messages. In one embodiment, private network 12 comprises a serial communication network with a serial multiplexor 20 interconnecting the server nodes 12 to the private network 18. In the event of a server or application failure, the surviving system 12, if appropriately configured, assumes the public network address of the failed system 12 and answer requests from clients 16 on network 14. In one embodiment, clients 16 perceive the failover process as a rapid reboot of the failed primary server.

Thus, ¶ 78 of Sreenivasan does not disclose or suggest that recovery modules of the type disclosed in Turek are configured to determine the status of a network node in accordance with a heartbeat messaging protocol as assumed incorrectly by the Examiner. Instead, ¶ 78 discloses in pertinent part that "Software running on each server 12 use private network 18 to exchange heartbeat and other control messages."

Paragraphs 111 and 112 of Sreenivasan read as follows:

> As noted above, the main component of the Cluster Membership Service is the Cluster Membership Daemon. In some embodiments, the Cluster Membership Daemon is responsible for running the whole protocol and is represented by the Membership Daemon that runs on it. The daemon maintains in an internal variable its current view of the Membership. The daemon is said to have delivered a new Membership when the value of that variable is changed.
>
> Each CMD sends messages to other CMD's by invoking a broadcast primitive. The destination of the broadcast are all the nodes in S except the originator. Typically, the broadcast primitive is the only way CMD sends messages. The semantic of the broadcast primitive are very weak. Message can be lost and there are little guarantees on the ordering at the receive end. Current implementation of the daemon uses UDP/IP, however any datagram transport can be substituted. The broadcast primitive prepends a header to the message. As stated above CMD uses one type of message. Each message contains useful information and at the same time can be considered as an "I'm alive message" from the sender. CMD is required to periodically broadcast a message. The interval between broadcasts is a configurable parameter.

Thus, neither ¶ 111 nor ¶ 112 of Sreenivasan teaches that recovery modules of the type disclosed in Turek are configured to determine the status of a network node in accordance with a heartbeat messaging protocol. Instead, the CMD processes running on each of the servers in the cluster communicate with each other to detect server failure. In particular, Sreenivasan teaches that each server 12 in the cluster runs Cluster Management Services (CMS) 32 and Group Communication Services (GCS) 34 (see ¶ 79), where an instance of the CMS service 12 is referred to as a Cluster Management Daemon (CMD) (see ¶ 85). Nodes are represented by the CMD processes that run on them and the failure of such a CMD is interpreted as the failure of

the node (see ¶ 85). The CMD processes running on the servers 12 communicate using a Cluster Management Protocol 36 that includes an initialization phase, a monitoring phase, and an agreement phase (see ¶¶ 85-88). During the monitoring phase, the nodes in the cluster send and receive heartbeat messages (see ¶ 89). Paragraphs 111 and 112 explain some of the details of the communications between the CMS processes running on the network nodes.

To summarize, the Examiner has taken the position that the CMD processes running on the servers of the Sreenivasan's cluster constitute "recovery modules." These CMD processes, however, are not mobile.

Therefore, both Turek and Sreenivasan fail to disclose or suggest software agents that determine the "status" of a network node, where the "status" is determinable in accordance with a heartbeat messaging protocol. Since the cited references do not disclose or suggest disclose or suggest all the limitations of the invention defined in claim 5, the rejection under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan should be withdrawn. See In re Lowry, 32 F.3d 1579, 1582, 32 USPQ2d 1 031, 1034 (Fed. Cir. 1994).

d. Appellant's rebuttal: the rationale given in support of the combination of Turek and Sreenivasan does not establish a *prima facie* case of obviousness

The rationale given by the Examiner in support of his proposed combination of Turek and Sreenivasan does not establish a *prima facie* case of obviousness. In particular, the Examiner's position is that it would have been obvious "to have incorporated the functionality of daemon (module or software agent) sending "I am alive message" (heartbeat messaging protocol) to determine the status of a network node as disclosed by Sreenivasan in the a system for managing a plurality of distributed nodes of a network as disclosed by Turek." In accordance with Turek's teachings, however, the management server 14 determines network errors or faults (see, e.g., col. 5, lines 31-60) and subsequently deploys the mobile software agents to diagnose and, if possible, correct the network error or fault (see, e.g., col. 5, lines 37-42). The incorporation of a heartbeat messaging protocol into the management server 14 to determine the status of a network node as disclosed by Sreenivasan (i.e., by exchanging heartbeat messages between software running on each server) would not result in the invention defined in claim 5, where "the recovery module is

configured to determine the status of a network node in accordance with a heartbeat messaging protocol."

For at least this additional reason, the rejection of claim 5 under 35 U.S.C. § 103(a) over Turek and Sreenivasan should be withdrawn.

e. Appellant's rebuttal: one skilled in the art at the time the invention was made would not have had any apparent reason to modify the teachings of Turek in the manner proposed by the Examiner

In accordance with Turek's teachings, the software agents are deployed only after an event, such as a network fault, has been determined by the management server 14 (see, e.g., col. 2, lines 35-37). Therefore, there is no need for the software agents to use a heartbeat messaging protocol to determine whether such an event originated from a particular node. Instead, each software agent need only be tailored to specifically identify the particular network fault that triggered the deployment of the software agent by the management server 14 (see, e.g., col. 5, lines 31-60 and col. 6, lines 23-59). As explained above, such information is not determinable in accordance with a heartbeat protocol due to the limited nature of the information conveyed by the heartbeat messages (i.e., "I'm here, are you here?") For at least this reason, one skilled in the art at the time the invention was made would not have had any apparent reason to modify the teachings of Turek in the manner proposed by the Examiner.

In addition, the Examiner has premised his proposed combination of Turek and Sreenivasan on the following rationale:

> It would have been obvious to one in the ordinary skill in the art at the time the invention was made to have incorporated the functionality of daemon (module or software agent) sending "I am alive message" (heartbeat messaging protocol) to determine the status of a network node as disclosed by Sreenivasan in the a system for managing a plurality of distributed nodes of a network as disclosed by Turek in order to make the managing system more reliable and responsive resulting in determining accurate diagnosis and status of the network nodes.

Neither Turek nor Sreenivasan, however, provides any support for the Examiner's assertion that modifying his proposed modification of Turek's system would "make the

managing system more reliable and responsive resulting in determining accurate diagnosis and status of the network nodes." This assertion is a pure fabrication of the Examiner's imagination.

Moreover, there is nothing in the disclosure of either Turek or Sreenivasan that would have led one skilled in the art at the time the invention was made to modify Turek's migratory software agents to run the server-node-based CMD processes (i.e., the instances of the Cluster Management Services (CMS) 32; see ¶ 85) that are disclosed in Sreenivasan. For example, each of the CMDs is designed to run on a single server of a cluster (see, e.g., ¶ 85). Neither Turek nor Sreenivasan nor the knowledge generally available even hints that such CMD processes could be incorporated into migratory software agents for the purpose of diagnosing and correcting errors or faults on the nodes to which the agents are deployed (see, e.g., col. 5, lines 41-47).

Even assuming only for the purposes of argument that Turek's migratory agents could be modified to run the CMD processes, one skilled in the art would not have had a reasonable basis for believing that the CMD framework would work when running on migratory software agents of the type described in Turek. In particular, the CMD framework is designed to operate with each of the CMDs running on a single respective server of a cluster. Neither Turek nor Sreenivasan provides any basis for believing that the CMD framework could work for its intended purpose if the CMDs were somehow able to migrate from one server to another as proposed by the Examiner.

In addition, instead of pointing to some teaching or suggestion in Turek, Sreenivasan, or the knowledge generally available to support the proposed combination of Turek and Harvell, the Examiner has relied on circular reasoning. In particular, the Examiner's proffered motivation (i.e., because it would "...make the managing system more reliable and responsive resulting in determining accurate diagnosis and status of the network nodes") assumes the result (i.e., the modification of Turek's system) to which the proffered "motivation" was supposed to have led one skilled in the art. Such circular reasoning cannot possibly support a rejection under 35 U.S.C. § 103(a). Indeed, such circular reasoning only evidences the fact that the Examiner improperly has engaged in impermissible hindsight reconstruction of the claimed invention, using applicants' disclosure as a blueprint for piecing together elements from the prior art in a manner that attempts to reconstruct the invention recited in claim 1 only with the benefit of impermissible hindsight (see KSR Int'l Co. v. Teleflex Inc., slip op. at 17: "A factfinder should

be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning."). The fact is that neither Turek nor Sreenivasan nor the knowledge generally available at the time the invention was made would have led one skilled in the art to believe that there was any problem to be solved or any advantage that would be gained by the Examiner's proposed modification of Turek's system.

Without any apparent reason for modifying Turek's system, the Examiner's rationale in support of the rejection of claim 5 amounts to no more than a conclusory statement that cannot support a rejection under 35 U.S.C. § 103.

Thus, contrary to the Examiner's position, one skilled in the art at the time the invention was made would not have been led to modify Turek's migratory software agents to run the CMD processes (i.e., the instances of the Cluster Management Services (CMS) 32; see ¶ 85) disclosed in Sreenivasan.

For at least these additional reasons, the rejection of claim 5 under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan should be withdrawn.

f. Conclusion

For the reasons explained above, the rejection of claim 5 under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan should be withdrawn.

3. Dependent claim 15

Claim 15 depends from independent claim 11 and recites that "the status of a network node is determined in accordance with a heartbeat messaging protocol."

Sreenivasan does not make-up for the failure of Turek to disclose or suggest the elements of independent claim 11 discussed above. Indeed, Sreenivasan does not disclose anything about recovery modules of the type disclosed in Turek, much less anything about such modules that are "configured to determine the status of a network node in accordance with a heartbeat messaging protocol." Moreover, one skilled in the art at the time the invention was made would not have had any apparent reason to combine Turek and Sreenivasan in the manner proposed by the Examiner for the reasons explained above in connection with independent claim 5.

For at least these reasons, the rejection of claim 15 under 35 U.S.C. § 103(a) over Turek in view of Sreenivasan should be withdrawn.

C. Rejection of claims 27-29 under 35 U.S.C. § 103(a) over Turek in view of Douik

The Examiner has rejected claims 27-29 under 35 U.S.C. § 103(a) over Turek in view of Douik (U.S. 6,012,152).

Each of claims 27-29 incorporates the features of independent claim 1. Douik does not make-up for the failure of Turek to teach or suggest the features of independent claim 1 discussed above. Therefore, claims 27-29 are patentable over Turek and Douik for at least the same reasons explained above in connection with independent claim 1.

Claims 27-29 also are patentable over Turek in view of Douik for at the following additional reasons.

1. Claim 27

Claim 27 recites that the network management module determines a number of the recovery modules needed to achieve a specified network monitoring service level, and launches the determined number of recovery modules into the network to achieve the specified network monitoring service level.

In support of the rejection of claim 27, the Examiner has stated that (see § 24 on pages 9-10 of the Office action dated March 20, 2008; emphasis added):

> ... Turek-Sreenivasan did not explicitly disclose, wherein the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and proactively launches the recovery modules into the network by transmitting respective ones of the recovery modules to the identified target network nodes. In the same field of endeavor Douik disclosed wherein the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and launches the recovery modules into the network by transmitting respective ones of the recovery modules to

> the identified target network nodes (col. 11, lines 64-67 & col. 12, lines 1-19).

As pointed out in the Amendment dated November 27, 2006 (see § IV.G.1 on pages 19-20), and again in the Response dated May 7, 2007 (see § III.C.1 on pages 25-26), and again in the Appeal Brief dated December 20, 2008 (see § VII.D.1 on pages 33-35), contrary to the Examiner's assumption, claim 27 does not recite that the network management module "identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and launches the recovery modules into the network by transmitting respective ones of the recovery modules to the identified target network nodes." Instead, claim 27 recites that "the network management module determines a number of the recovery modules needed to achieve a specified network monitoring service level, and launches the determined number of recovery modules into the network to achieve the specified network monitoring service level." Thus, on its face, the Examiner's rejection of claim 27 does not establish a *prima facie* case of obviousness (see MPEP § 706.02(j)).

Moreover, Douik does not teach or suggest anything about migratory recovery modules, much less anything about determining a number of the recovery modules needed to achieve a specified network monitoring service level and launching the determined number of recovery modules into the network to achieve the specified network monitoring service level.

The disclosure on which the Examiner's rejection of claim 27 is premised reads as follows (i.e., col. 11, line 64 - col. 12, line 19):

> In yet another aspect, the present invention is a method of proactively managing software faults in a mobile telecommunications network. The method begins by storing knowledge in a knowledge base, the knowledge including a functional model of the network, fault models, and fault scenarios; monitoring the network for observed events and symptoms; and determining a suspected fault to explain the observed events and symptoms, the determining step comprising comparing the observed events and symptoms with stored performance data and statistics, and analyzing the comparison with the stored knowledge. This is followed by determining whether the suspected fault is a known fault; implementing a preventive solution upon determining that the suspected fault is a known fault; and performing a fault

> trend analysis upon determining that the suspected fault is not a known fault. This is followed by performing diagnostic tests; determining whether a successful diagnosis was obtained; performing a fault localization process upon determining that a successful diagnosis was obtained, the fault localization process including analyzing relationships between components involved in the diagnosis of the fault; and providing diagnosis and localization information to trouble shooters.

This disclosure does not describe anything whatsoever about migratory recovery modules, much less anything about determining a number of the recovery modules needed to achieve a specified network monitoring service level and launching the determined number of recovery modules into the network to achieve the specified network monitoring service level.

For at least these additional reasons, the Examiner's rejection of claim 27 under 35 U.S.C. § 103(a) over Turek in view of Douik should be withdrawn.

2. Claim 28

Claim 28 recites that the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and launches the recovery modules into the network by transmitting respective ones of the recovery modules to the identified target network nodes.

In support of the rejection of claim 28, the Examiner has stated that (see § 24 on pages 9-10 of the Office action dated March 20, 2008; emphasis added):

> ... Turek-Sreenivasan did not explicitly disclose, wherein the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and proactively launches the recovery modules into the network by transmitting respective ones of the recovery modules to the identified target network nodes. In the same field of endeavor Douik disclosed wherein the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and launches the recovery modules into the network by transmitting respective ones of the recovery modules to

> the identified target network nodes (col. 11, lines 64-67 & col. 12, lines 1-19).

The disclosure on which the Examiner's rejection of claim 28 is premised reads as follows (i.e., col. 11, line 64 - col. 12, line 19):

> In yet another aspect, the present invention is a method of proactively managing software faults in a mobile telecommunications network. The method begins by storing knowledge in a knowledge base, the knowledge including a functional model of the network, fault models, and fault scenarios; monitoring the network for observed events and symptoms; and determining a suspected fault to explain the observed events and symptoms, the determining step comprising comparing the observed events and symptoms with stored performance data and statistics, and analyzing the comparison with the stored knowledge. This is followed by determining whether the suspected fault is a known fault; implementing a preventive solution upon determining that the suspected fault is a known fault; and performing a fault trend analysis upon determining that the suspected fault is not a known fault. This is followed by performing diagnostic tests; determining whether a successful diagnosis was obtained; performing a fault localization process upon determining that a successful diagnosis was obtained, the fault localization process including analyzing relationships between components involved in the diagnosis of the fault; and providing diagnosis and localization information to trouble shooters.

In this disclosure, Douik merely compares observed events to stored performance data and statistics in order to determine a suspected fault to explain the observed events and symptoms. Douik does not even hint that target nodes are identified statistically to achieve a specified confidence level of network monitoring reliability. Moreover, Douik does not teach or suggest anything about migratory recovery modules and launching the recovery modules into the network by transmitting respective ones of the recovery modules to the identified target network nodes.

For at least this additional reason, the Examiner's rejection of claim 28 under 35 U.S.C. § 103(a) over Turek in view of Douik should be withdrawn.

3. Claim 29

Claim 29 depends from claim 11 and recites: determining a number of the recovery modules needed to achieve a specified network monitoring service level; statistically identifying target ones of the network nodes to achieve a specified confidence level of network monitoring reliability; and transmitting the determined number of the recovery modules to the identified target network nodes.

Claim 29 recites features that essentially track the pertinent features of claim 28 discussed above. Therefore, claim 29 is patentable over Turek in view of Douik for at least the same reasons explained above in connection with claim 28.

VIII. Conclusion

For the reasons explained above, all of the pending claims are now in condition for allowance and should be allowed.

Charge any excess fees or apply any credits to Deposit Account No. 08-2025.

Respectfully submitted,

Date: December 18, 2008

/Edouard Garcia, Reg. No. 38,461/
Edouard Garcia
Reg. No. 38,461
Telephone No.: (650) 965-8342

Please direct all correspondence to:

Hewlett-Packard Company
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, CO 80528-9599

CLAIMS APPENDIX

The claims that are the subject of Appeal are presented below.

Claim 1 (previously presented): A system for managing a plurality of distributed nodes of a network, comprising:

a network management module that launches migratory recovery modules into the network to monitor status of each of the network nodes;

wherein each of the recovery modules is configured to migrate from one network node to another, determine a respective status of each of the network nodes to which it has migrated, and initiate a recovery process on ones of the network nodes having one or more failed node processes, the recovery modules determine the status of each of the network nodes, and the network management module monitors transmissions that are received from the recovery modules to provide periodic monitoring of the status of each of the network nodes.

Claim 2 (previously presented): The system of claim 1, wherein at least one of the recovery modules comprises a respective routing component for determining next hop addresses for migrating the recovery module from an origin network node to a series of successive destination network nodes.

Claim 3 (previously presented): The system of claim 2, wherein the routing component is configured to determine the next hop addresses based upon a routing table stored at the origin network node.

Claim 4 (previously presented): The system of claim 1, wherein at least one of the recovery modules is configured to determine the status of a network node by sending an inter-process communication to a node process.

Claim 5 (previously presented): A system for managing a plurality of distributed nodes of a network, comprising:

a recovery module configured to migrate from one network node to another, determine a status of a network node, and initiate a recovery process on a network node having one or more failed node processes, wherein the recovery module is configured to determine the status of a network node in accordance with a heartbeat messaging protocol.

Claim 6 (previously presented): The system of claim 1, wherein each of the recovery modules is configured to initiate a recovery process on a network node having one or more failed node processes in accordance with a restart protocol.

Claim 7 (previously presented): The system of claim 6, wherein each of the recovery modules is configured to initiate a restart of a failed node process by transmitting a request to a process execution service operating on the failed network node.

Claim 8 (previously presented): The system of claim 1, wherein each of the recovery modules is configured to transmit a respective node status message to the network management module.

Claim 9 (previously presented): The system of claim 8, wherein each of the node status messages comprises information obtained from a respective log file generated at a respective one of the network nodes having one or more failed node processes.

Claim 10 (canceled)

Claim 11 (previously presented): A method for managing a plurality of distributed nodes of a network, comprising:

(a) on a current one of the network nodes, determining a status of the current network node;

(b) in response to a determination that the current network node has one or more failed node processes, initiating a recovery process on the current network node;

(c) after initiating the recovery process, migrating from the current network node to a successive one of the network nodes; and

(d) repeating (a), (b), and (c) with the current network node corresponding to the successive network node for each of the nodes in the network.

Claim 12 (original): The method of claim 11, wherein migrating from one network node to another comprises determining a next hop address from an origin network node to a destination network node.

Claim 13 (original): The method of claim 12, wherein the next hop address is determined based upon a routing table stored at the origin network node.

Claim 14 (original): The method of claim 11, wherein the status of a network node is determined by sending an inter-process communication to a node process.

Claim 15 (original): The method of claim 11, wherein the status of a network node is determined in accordance with a heartbeat messaging protocol.

Claim 16 (previously presented): The method of claim 11, wherein a recovery process is initiated on a network node having one or more failed node processes in accordance with a restart protocol.

Claim 17 (original): The method of claim 16, wherein a restart of a failed node process is initiated by transmitting a request to a process execution service operating on the failed network node.

Claim 18 (original): The method of claim 11, further comprising transmitting a node status message to a network management module operating at a network management network node.

Claim 19 (previously presented): The method of claim 11, further comprising launching into the network a plurality of recovery modules, each configured to migrate from one network node to another, determine the status of a network node, and initiate a recovery process on a failed network node having one or more failed node processes.

Claim 20 (previously presented): A computer program for managing a plurality of distributed nodes of a network, the computer program residing on a computer-readable medium and comprising computer-readable instructions for causing a computer to perform operations comprising:

migrating the computer program from one network node to a series of successive network nodes;

determining a status of a current one of the network nodes to which the computer program has migrated;

in response to a determination that the current network has one or more failed node processes, initiating a recovery process on the current network node; and

after initiating the recovery process on the current network node, migrating from the current network node to a successive one of the network nodes.

Claim 21 (previously presented): The system of claim 1, wherein each of the recovery modules is a software object that is instantiatable by a respective operating environment on each network node.

Claim 22 (previously presented): The system of claim 21, wherein the operating environment on each of the network nodes provides each of the recovery modules with access to status monitoring resources, recovery resources, and native operative system resources that are available at each of the network nodes.

Claim 23 (previously presented): The system of claim 1, wherein, upon migrating from a first one of the network nodes to a second one of the network nodes and being instantiated on the second node, each of the recovery modules determines a status of the second network node.

Claim 24 (previously presented): The system of claim 23, wherein each of the recovery modules initiates the recovery process on the second node in response to a determination that the second node has one or more failed node processes.

Claim 25 (previously presented): The system of claim 23, wherein each of the recovery modules is configured to migrate to a third one of the network nodes after determining the status of the second network node.

Claim 26 (canceled)

Claim 27 (previously presented): The system of claim 1, wherein the network management module determines a number of the recovery modules needed to achieve a specified

network monitoring service level, and launches the determined number of recovery modules into the network to achieve the specified network monitoring service level.

Claim 28 (previously presented): The system of claim 1, wherein the network management module statistically identifies target ones of the network nodes to achieve a specified confidence level of network monitoring reliability, and launches the recovery modules into the network by transmitting respective ones of the recovery modules to the identified target network nodes.

Claim 29 (previously presented): The method of claim 11, further comprising:

determining a number of the recovery modules needed to achieve a specified network monitoring service level;

statistically identifying target ones of the network nodes to achieve a specified confidence level of network monitoring reliability; and

transmitting the determined number of the recovery modules to the identified target network nodes.

Claim 30 (previously presented): The system of claim 1, wherein the network management module monitors number of network node failures reported by the recovery modules and launches more migratory recovery modules into the network as the number of reported failures increases.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to 37 CFR §§ 1.130, 1.131, or 1.132 or any other evidence entered by the Examiner and relied upon by Appellant in the pending appeal. Therefore, no copies are required under 37 CFR § 41.37(c)(1)(ix) in the pending appeal.

RELATED PROCEEDINGS APPENDIX

Appellant is not aware of any decisions rendered by a court or the Board that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal. Therefore, no copies are required under 37 CFR § 41.37(c)(1)(x) in the pending appeal.